Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 634848, 11 pages
http://dx.doi.org/10.1155/2014/634848

Research Article

Fractal Image Coding Based on a Fitting Surface

Sheng Bi and Qiang Wang

College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Qiang Wang; wangqiang2011@dlmu.edu.cn

Received 5 June 2014; Revised 5 August 2014; Accepted 6 August 2014; Published 21 August 2014

Academic Editor: Filomena Cianciaruso

A no-search fractal image coding method based on a fitting surface is proposed. In our research, an improved gray-level transform with a fitting surface is introduced. One advantage of this method is that the fitting surface is used for both the range and domain blocks and one set of parameters can be saved. Another advantage is that the fitting surface can approximate the range and domain blocks better than the previous fitting planes; this can result in smaller block matching errors and better decoded image quality. Since the no-search and quadtree techniques are adopted, smaller matching errors also imply less number of blocks matching which results in a faster encoding process. Moreover, by combining all the fitting surfaces, a fitting surface image (FSI) is also proposed to speed up the fractal decoding. Experiments show that our proposed method can yield superior performance over the other three methods. Relative to range-averaged image, FSI can provide faster fractal decoding process. Finally, by combining the proposed fractal coding method with JPEG, a hybrid coding method is designed which can provide higher PSNR than JPEG while maintaining the same Bpp.

1. Introduction

The basic idea of fractal image coding was first proposed by Barnsley [1]. Its kernel issue is to find an iterated function system whose fixed point can approximate the input image well. Then, Jacquin [2] introduced the first practical fractal coding scheme. After many years of development, it has been successfully used in many image processing applications such as image compression [3], image denoising [4, 5], image retrieval [6, 7], image magnification [8, 9], and image watermarking [10–12].

Although the fractal coding method has the advantage of potential high compression ratio, resolution independence, and fast decoding, its main drawback is very time-consuming in the encoding process. In order to reduce the computational complexity of fractal encoding, converting the global search to local search is an effective way to solve this problem. It mainly consists of classification techniques and feature vector techniques. For the former ones, the range blocks and domain blocks are first divided into different categories and the block matching is only carried out within the same category [13, 14]. For the latter ones, the block matching process is carried out in the feature space. Due to the lower dimension of features and more effective searching strategies, such as kd-tree method, the fractal encoding process can be finished in a short time [15–17]. Typically, Furao and Hasegawa [18] introduced the first no-search fractal encoding algorithm which can provide faster encoding process and higher compression ratio at the expense of poor quality of the decoded image. Afterwards, Wang et al. [19, 20] proposed the adaptive plane and fitting plane to improve the quality of the decoded image, respectively. In particular, the latter one can achieve higher compression ratio, better decoded image quality, and shorter encoding time than the previous similar methods. In our research, we propose an improved gray-level transformation whose fitting surface is used to fit both the range blocks and domain blocks. Moreover, the fitting surface in our method can approximate the range block better than Wang's fitting plane. Due to the no-search and quadtree method adopted, the proposed method can also maintain a fast encoding speed. Moreover, an initial image for fast fractal decoding is also proposed. Generally, accelerating the fractal decoding can be realized by two ways: one is to adopt improved iteration strategies [21–23], such as the one buffer decoding method [23]. The other one is to select an initial image which can approximate the original

image well. Moon et al. [24] proposed an approximated range-averaged image as the initial image and the range-averaged image is considered as the ideal initial image. In our research, we propose a novel initial image which can provide faster decoding speed. Since the proposed fitting surface itself is more similar to the corresponding range block, all the fitting surfaces can constitute a better fitting surface image (FSI) as the initial image in fractal decoding process without extra computations. Experimental results show that the proposed no-search fractal encoding method can provide better performance than Jacquin [2], Tong [25], and Wang's methods [20]. Moreover, compared with the range-averaged image, FSI can also provide faster decoding speed. Finally, by combining our proposed fractal encoding method with JPEG, we design a hybrid coding method which can provide higher PSNR than JPEG while maintaining the same Bpp.

This paper is organized as follows. In Section 2, we describe the conventional fractal image coding method and the corresponding improved gray-level transforms. In Section 3, an improved gray-level transform with a second order fitting surface is proposed. In Section 4, a novel initial image is introduced to accelerate the fractal decoding process. In Section 5, the experimental results, such as Bpp, encoding time, and decoded image quality, are given and analyzed in detail. The final conclusion is presented in Section 6.

2. Review of Fractal Image Coding

According to the Collage theorem, for an input image $\mathbf{F}$, the fractal image encoding process is to construct an iteration function system W which can be used to reconstruct the original image approximately. The reconstruction procedures can be described as follows:

$$W^{(k)}(\mathbf{F}_0) \Longrightarrow \overline{\mathbf{F}}, \quad k \longrightarrow \infty, \tag{1}$$

where k indicates the kth iteration. $\mathbf{F}_0$ is the initial image. $\overline{\mathbf{F}}$ denotes the fixed point of W which is an approximated image of $\mathbf{F}$.

In the practical implementation of the fractal encoding process, the input image is firstly partitioned into nonoverlapping range blocks. The domain blocks are obtained by sliding a square window throughout the original image. Commonly, the size of the domain blocks is twice the size of the range blocks. For each range block, the corresponding best matching domain block can be found in a domain block pool. The domain block pool is built by contracting all the domain blocks to the same size of the range blocks. Furthermore, the domain block pool is extended by eight isometric transformations which contain the identity, rotation through 90°, rotation through 180°, rotation through 270°, reflection about the middle vertical axis, reflection about the middle horizontal axis, reflection about the first diagonal, and reflection about the second diagonal. Figure 1 illustrates the above eight isometric transformations in order. In order to reduce the matching error, each contracted domain block should be transformed by the following gray-level transform:

$$\varphi(\mathbf{D}) = \alpha\mathbf{D} + \beta\mathbf{I}, \tag{2}$$

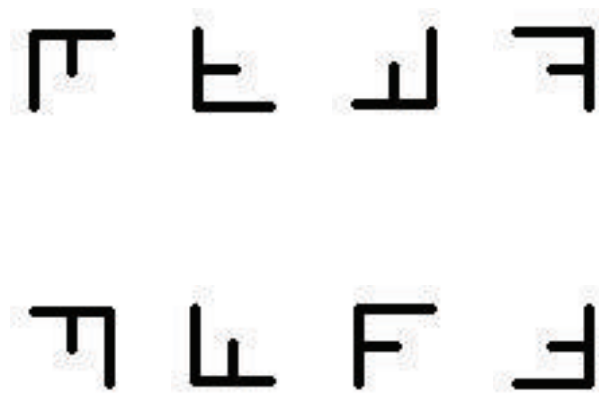


Figure 1: Eight isometric transformations in fractal encoding.

where α and β denote the scaling coefficient and offset coefficient, respectively. For each range block $\mathbf{R}$, the best matching domain block can be obtained by minimizing the following equation:

$$\text{Error} = \min_{\alpha,\beta} \left\| \mathbf{R} - \alpha\mathbf{D} - \beta\mathbf{I} \right\|^2, \tag{3}$$

where $\|\bullet\|$ is the two-norm. $\mathbf{I}$ denotes a matrix whose elements are all ones. Error is the minimum matching error for each range block. With the least squares method, α and β can be computed as follows:

$$\alpha = \frac{\left\langle \mathbf{R} - \overline{r}\mathbf{I}, \mathbf{D} - \overline{d}\mathbf{I} \right\rangle}{\left\| \mathbf{D} - \overline{d}\mathbf{I} \right\|^2}, \qquad \beta = \overline{r} - \alpha\overline{d}, \tag{4}$$

where $\langle \bullet, \bullet \rangle$ is the inner product. $\overline{d}$ and $\overline{r}$ are the mean values of the contracted domain block and the range block, respectively. The fractal encoding process can be finished by storing all the transformation parameters as the fractal code.

At the fractal decoding phase, the transformation parameters are recursively applied to any starting image and an iteration image sequence is constructed. The image sequence converges after about ten iterations. By selecting the commonly used black image as the initial image, Figure 2 illustrates the first six iteration images for Barbara image.

Different from (2), Tong and Pi [25] proposed a modified gray-level transform as follows:

$$\varphi'(\mathbf{D}) = s\left(\mathbf{D} - \overline{d}\mathbf{I}\right) + \overline{r}\mathbf{I}. \tag{5}$$

The corresponding minimum matching error function can be described as

$$E(\mathbf{R}, \mathbf{D})^2 = \min_i \left\| s_i(\mathbf{D} - \overline{d}\mathbf{I}) - (\mathbf{R} - \overline{r}\mathbf{I}) \right\|^2, \tag{6}$$

where s_i is the scaling coefficient. $\overline{d}$ and $\overline{r}$ are also the mean values of the contracted domain block and the range block, respectively. The main advantage of this method is that we only need to search the optimal coefficient s_i and it is more efficient than the previous method in (2). In order to reduce the minimum matching error further, Wang et al. [20] proposed an improved gray-level transform as follows:

$$\varphi''(\mathbf{D}) = s\left(\mathbf{D} - \mathbf{P}_{\mathbf{D}}\right) + \mathbf{P}_{\mathbf{R}}. \tag{7}$$

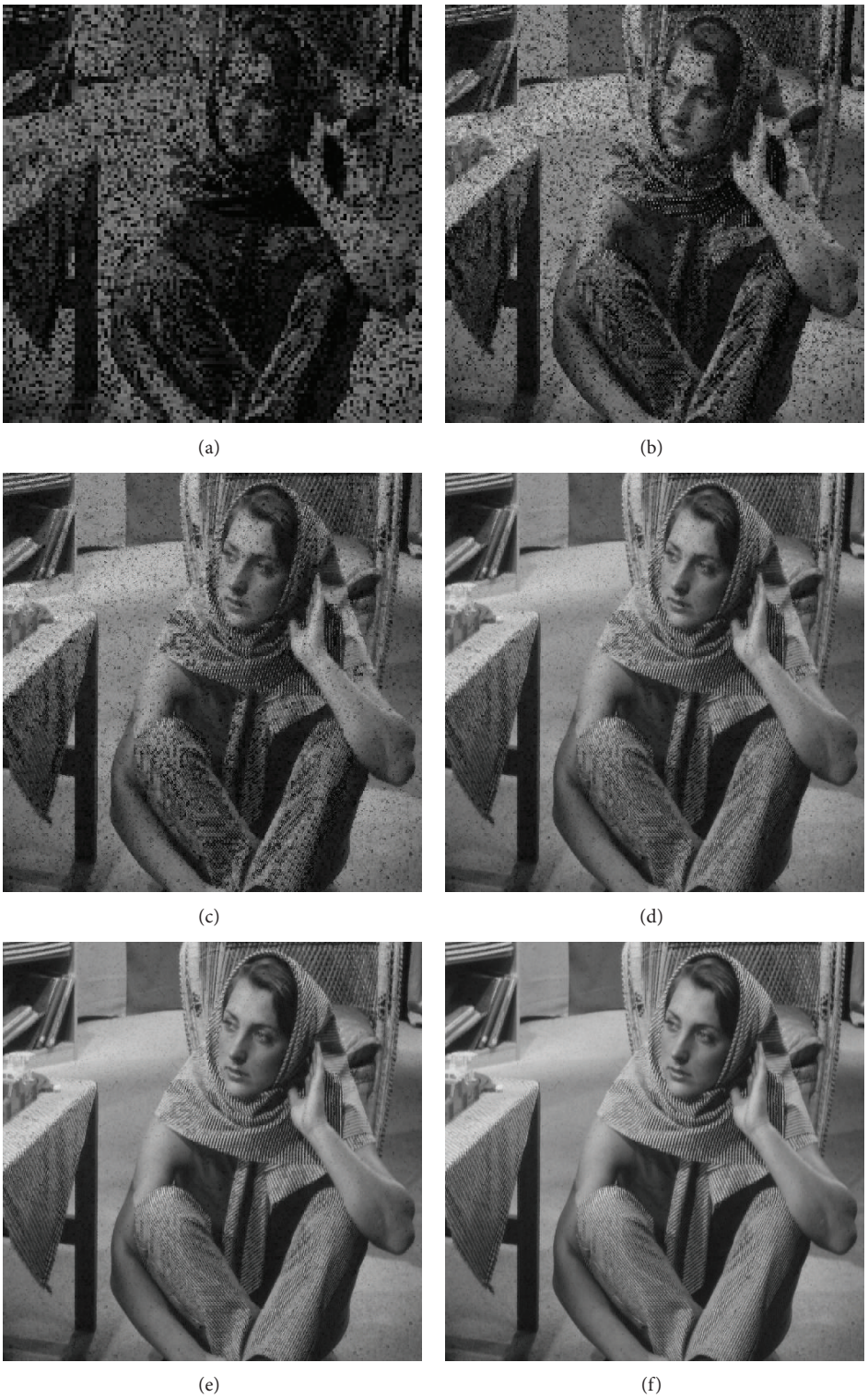


Figure 2: The first six iterations for Barbara image in fractal decoding.

The corresponding minimum matching error function can be described as

$$E(\mathbf{R},\mathbf{D})^2 = \min_i \left\| s_i(\mathbf{D}-\mathbf{P_D}) - (\mathbf{R}-\mathbf{P_R}) \right\|^2, \tag{8}$$

where $\mathbf{P_R}$ and $\mathbf{P_D}$ are the fitting planes of the range block $\mathbf{R}$ and the contracted domain block $\mathbf{D}$, respectively. The fitting plane can be described as follows:

$$z = ax + by + c, \quad x, y = 1, 2, \ldots, L, \tag{9}$$

where L is the width or height of the range blocks. x and y are the variables to represent the position of each pixel. z is the pixel gray value. a, b, and c are the parameters to be estimated.

3. An Improved Gray-Level Transform Using a Fitting Surface

In this section, a second order surface instead of a first order one in (9) is proposed which can be represented as follows:

$$z = af^2(x) + bf(x) + cf^2(y) + df(y) + e \qquad x, y = 1, 2, \ldots, L, \tag{10}$$

where

$$f(x) = \begin{cases} x - \dfrac{L}{2} & \dfrac{L}{2} < x \le L \\ x - \dfrac{L}{2} - 1 & 1 \le x \le \dfrac{L}{2}. \end{cases} \tag{11}$$

For arbitrary range block $\mathbf{R}$, we need to find the coefficients a, b, c, d, and e that can make the surface best fit the range block $\mathbf{R}$. We can solve this problem by the least squares method:

$$E = \min_{a,b,c,d,e} \sum_{x=1}^{L} \sum_{y=1}^{L} (\mathbf{R}(x, y) - z)^2, \tag{12}$$

where z is pixel gray value in (10). Set the derivatives of (12) with respect to a, b, c, d, and e to zero, respectively. By solving the system of equations, we can obtain the optimal coefficients as follows:

$$\begin{aligned} a &= \left(-\left(60\left(\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)(L+1)(L+2) - 12\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)f^2(x)\right)\right)\right) \\ &\quad \times \left(\left(L^2(4L+11)(L-2)(L+2)(L+1)\right)\right)^{-1}, \\ b &= \frac{12\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)f(x)}{L^2(L+2)(L+1)}, \\ c &= \left(-\left(60\left(\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)(L+1)(L+2) - 12\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)f^2(y)\right)\right)\right) \\ &\quad \times \left(\left(L^2(4L+11)(L-2)(L+2)(L+1)\right)\right)^{-1}, \\ d &= \frac{12\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)f(y)}{L^2(L+2)(L+1)}, \\ e &= \left(\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)\left(14L^2+33L-2\right) - 60\left(\sum_{x=1}^{L}\sum_{y=1}^{L}\mathbf{R}(x,y)\left(f^2(x)+f^2(y)\right)\right)\right) \\ &\quad \times \left(L^2(4L+11)(L-2)\right)^{-1}. \end{aligned} \tag{13}$$

Figure 3(a) illustrates an original surface which can be considered as a range block. Figures 3(b) and 3(c) are Wang's fitting plane and the proposed fitting surface, respectively. We can observe that the proposed fitting surface approximates the original surface better than the fitting plane. Substantially, the mean squared errors of Wang's fitting plane and the proposed fitting surface are 300 and 200, respectively.

Furthermore, we propose an improved gray-level transform by making use of the surface stated in (10) as follows:

$$\varphi'''(\mathbf{D}) = s(\mathbf{D} - \mathbf{S_R}) + \mathbf{S_R}. \tag{14}$$

The corresponding minimum matching error function can be denoted as

$$E(\mathbf{R}, \mathbf{D})^2 = \min_i \left\|(\mathbf{R} - \mathbf{S_R}) - s_i(\mathbf{D} - \mathbf{S_R})\right\|^2, \tag{15}$$

where $\mathbf{S_R}$ is the fitting surface of the range block $\mathbf{R}$ and both the range block $\mathbf{R}$ and the domain block $\mathbf{D}$ adopt $\mathbf{S_R}$ as their fitting surface. On one hand, we do not need to use another surface to represent the domain block $\mathbf{D}$ and one set of parameters can be saved to improve the compression ratio. On the other hand, we suppose that, for arbitrary range block $\mathbf{R}$, the best matching domain block $\mathbf{D}$ is very similar to the range block $\mathbf{R}$ and can also provide minor matching error with the same fitting surface $\mathbf{S_R}$. In order to check the effectiveness of the proposed gray-level transform, in Figure 4(a), we select one hundred 8×8 range blocks randomly in Barbara image. Figure 4(b) illustrates their matching errors and we can see that, for most of the range blocks, our proposed method can provide smaller matching errors than Wang's method [20]. In addition, the minimum matching error function in (15) can be also described as follows:

$$E(\mathbf{R}, \mathbf{D})^2 = \left\|\mathbf{R} - \mathbf{S_R}\right\|^2 - s^2\left\|\mathbf{D} - \mathbf{S_R}\right\|^2 \le \left\|\mathbf{R} - \mathbf{S_R}\right\|^2. \tag{16}$$

From (16), we know that, since the proposed fitting surface $\mathbf{S_R}$ can approximate the range block $\mathbf{R}$ better than the fitting plane, smaller matching errors can be achieved in our method. This is consistent with Figure 4(b).

4. A Novel Initial Image for Fast Decoding

At the decoding phase of our proposed method, in each level of the quadtree structure established in the encoding process, all the range blocks will be reconstructed by the same way as (14). One iteration is finished by combining all the reconstructed range blocks with different sizes. $\mathbf{S_R}$ can be computed off line by (10) and all the fitting surfaces will be used repeatedly. Since (14) is invariable in the decoding process,

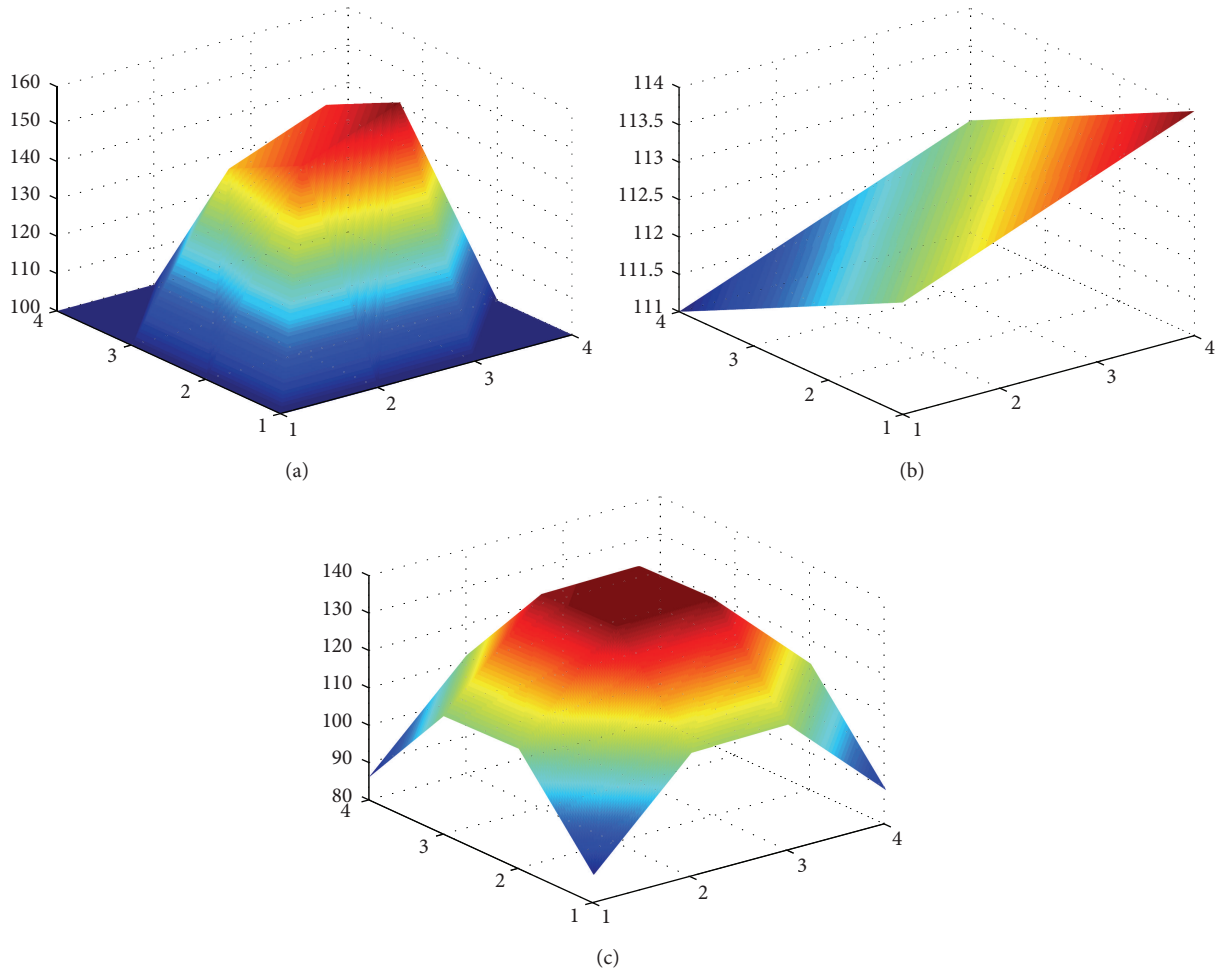


Figure 3: Comparison between the fitting plane and the fitting surface. (a) Original surface. (b) Fitting plane. (c) Fitting surface.

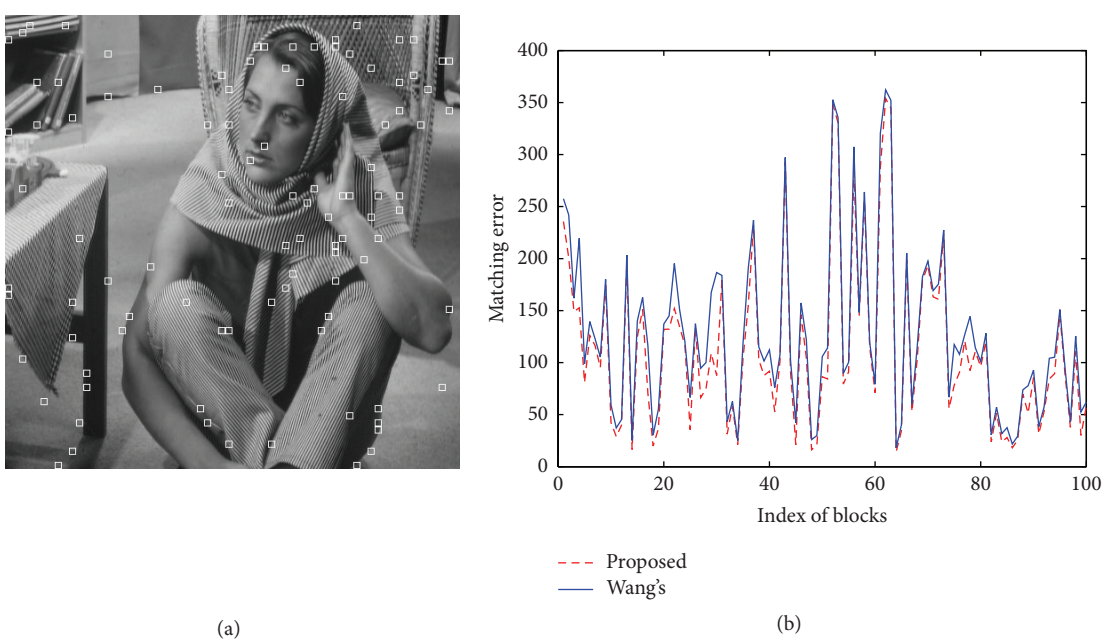
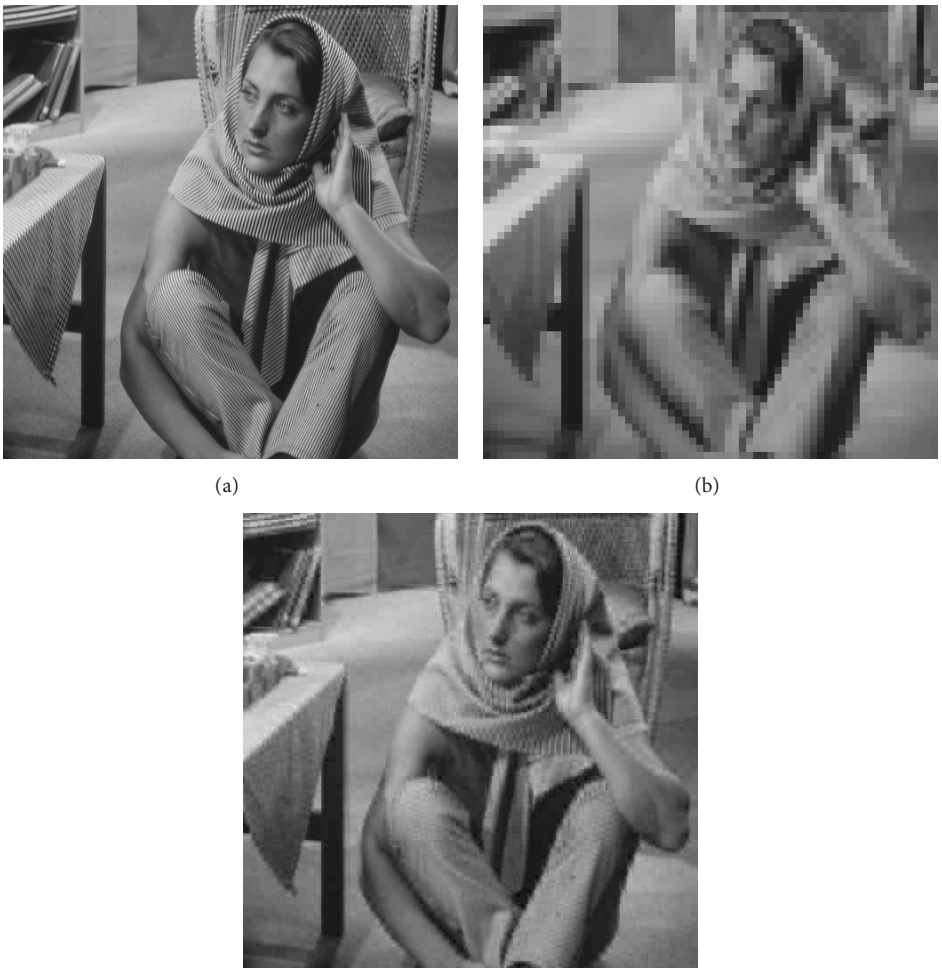


Figure 4: Matching error comparison between the proposed method and Wang's method. (a) 100 range blocks of size 8×8 selected randomly in Barbara image marked by "□." (b) Matching error comparison for the 100 range blocks.



(a) (b) (c)

Figure 5: Comparison between Moon's range-averaged image and FSI with respect to the original image. (a) Original image. (b) Range-averaged image. (c) FSI.

once the iteration strategy is specified, the convergence speed of the decoding process depends heavily on the initial image. The initial image is more similar to the input image; the reconstructed image in each iteration will approximate the decoded image better. In our research, instead of the initial black image which is commonly used, we proposed an initial image represented as follows:

$$\mathrm{FSI} = \bigcup_i \left(\mathbf{S}_{\mathbf{R}}^i\right), \tag{17}$$

where $\mathbf{S}_{\mathbf{R}}^i$ is the ith fitting surface, $\bigcup(\bullet)$ denotes the seamless splicing operation, and an initial image called fitting surface image (FSI) can be constituted by all the $\mathbf{S}_{\mathbf{R}}$s with the same order as corresponding range blocks. Since all the $\mathbf{S}_{\mathbf{R}}$s can be computed off line, we can get FSI without extra computations. By Moon's opinion [24], the range-averaged image is considered as the ideal initial image. The range-averaged image is combined with the blocks whose pixel intensities are all the mean values of the corresponding range blocks. Compared with the fitting surface we proposed, the blocks in range-averaged image can be only regarded as the simplest fitting surface. Thus, the proposed fitting surface in (10) can approximate arbitrary range block better than mean value block, and analogously the proposed FSI will be more similar to the original image than the range-averaged image. If the size of range blocks is set to be 8×8, Figures 5(a), 5(b), and 5(c) illustrate the original image, range-averaged image, and FSI for Barbara image, respectively. We can observe that, compared with the range-averaged image, FSI can preserve more details and approximate the original image better.

5. Experiments

The 512×512 Barbara image is used in our experiment. The scaling coefficient s_i is quantized by two bits and they are 0, 0.25, 0.5, and 0.75. The parameters a, b, c, d, and e for the fitting surface are quantized by 4, 4, 4, 4, and 6 bits, respectively. The experiments are carried out on a Pentium Dual-Core 2.93 GHz PC and programmed using MATLAB software. In order to estimate the performance of our proposed method, we will compare our algorithm with Jacquin [2], Tong [25], and Wang's methods [20] by the

encoding time, the quality of the decoded image, and Bpp. The decoded image quality is measured by peak signal to noise ratio (PSNR):

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\left((1/MN)\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{f}_{ij}-\mathbf{f}_{ij}^{*}\right)^2\right)}\right), \tag{18}$$

where M and N are the height and width of the input image. $\mathbf{f}$ and $\mathbf{f}^*$ represent the original image and the decoded image, respectively.

5.1. Compare the Proposed Method with the Other Three Methods. In this part, the no-search and quadtree method will be incorporated into the fractal coding method. The quadtree structure we use contains four levels and the sizes of range blocks are 16×16, 8×8, 4×4, and 2×2, respectively. In the encoding process, the input image is firstly divided into nonoverlapping range blocks of 16×16 pixels. We encode the range blocks one by one with the fractal encoding method and compare the matching error with a given error threshold T_i, $i = 1, 2, 3$, which changes by $T_{i+1} = 2 \times T_i$ as the quadtree level increases. Whether the range blocks should be divided into four subblocks is determined by whether the matching error is larger than the threshold. The above procedures will be repeated level by level until the size of range blocks is 2×2. We summarize the above procedures as follows:

Step 1. Set an error threshold T_1 and divide the input image into range blocks of 16×16 pixels. Assign extra two bits for each range block which are used to label the quadtree level it belongs to.

Step 2. For each range block, as illustrated in Figure 6, appoint the domain block as the best matching domain block directly which has the same center with the range block, compute the corresponding coefficients a, b, c, d, and e with (13), search the best scaling coefficient s_k, and obtain the minimum matching error by (15). If the minimum matching error is less than T_i, store the corresponding coefficients a, b, c, d, e, and s_k as the fractal code and label them. If not, divide the range block into four subblocks uniformly and insert them into the next level.

Step 3. Repeat Step 2 until the size of range blocks is 2×2. Encode them with the gray-level transform in (5) and store all the coefficients as the fractal code.

We compare the proposed method with the other three methods under the same environment and the main differences are the gray-level transforms in (2), (5), (7), and (14). The encoding speed is mainly determined by the total number of blocks encoded in four quadtree levels. Smaller number of blocks matching can result in shorter encoding time. From the last column of Table 1, it is clearly seen that, for each T_1, the proposed method needs to encode the least number of range blocks and this results in the shortest encoding time which is shown in the last row of Table 2. The number of encoded blocks increases from the proposed method to Wang's method to Jacquin and Tong's methods. Jacquin and Tong's methods need to encode the most range blocks and this will result in longer encoding time which is also shown in the eleventh and twelfth rows of Table 2. We also can observe that Jacquin and Tong's methods perform almost the same number of blocks matching. Furthermore, in each block matching, the parameters in Jacquin's method are computed by (4) which leads to more computations than Tong's method. Thus, the encoding time of Jacquin's method is longer than Tong's method.

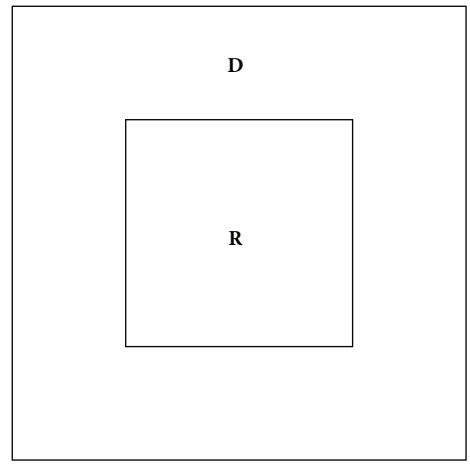


Figure 6: Relative position of the range block and its corresponding domain block.

From Figures 3 and 4, we know that the fitting surface $\mathbf{S_R}$ can approximate the corresponding range block $\mathbf{R}$ better; the corresponding matching error $E(\mathbf{R}, \mathbf{D})$ can achieve a smaller value than Wang's method. Consequently, for a given threshold T_1, more range blocks will be encoded in the quadtree level with larger block size. From Table 1, we can see that, from Jacquin and Tong's methods to Wang's method to our method, the number of blocks with larger size increases and the number of blocks with smaller size decreases. The number of 16×16 blocks in our method is always larger than the other three methods. On the one hand, smaller matching errors in the encoding process can result in higher quality of the decoded image which is listed in the sixth row of Table 2. On the other hand, larger number of blocks with larger size will result in improving Bpp. Figure 7(a) also illustrates the PSNR versus Bpp for the four methods; we can see that the proposed method can obtain the highest decoded image quality all the time with different Bpps. If Bpp is larger than 1.2, the PSNR of Wang's method is slightly smaller than Tong's method. If not, Wang's method can obtain higher PSNR than Tong's method. Figure 7(b) illustrates the PSNR versus encoding time for the four methods. We can see that, under the same encoding time, our proposed method can provide the highest PSNRs than the other three methods. Figure 8 illustrates the decoded Barbara image with $T_1 = 11$. Figure 9 illustrates the quadtree structure for Barbara image with $T_1 = 11$. In summary, compared with the other three methods, the proposed method can provide higher PSNR at the same Bpp and higher PSNR at the same encoding time.

Table 1: Number of blocks in different quadtree levels for the four methods.

Image	T_1	Method	Level 1 (16 × 16)	Level 2 (8 × 8)	Level 3 (4 × 4)	Level 4 (2 × 2)	Total
Barbara	7	Jacquin's	277	1394	4826	6200	12697
		Tong's	277	1394	4826	6200	12697
		Wang's	313	1383	4540	5216	11452
		Proposed	327	1442	4571	3252	9592
	11	Jacquin's	363	1734	3430	840	6367
		Tong's	363	1735	3426	840	6364
		Wang's	401	1678	3134	488	5701
		Proposed	417	1687	2927	148	5179
	15	Jacquin's	450	1862	1735	4	4051
		Tong's	450	1862	1735	4	4051
		Wang's	488	1769	1499	4	3760
		Proposed	535	1605	1403	4	3547

Table 2: Performance comparison between the proposed method and the other three methods.

Image		Barbara		
T		7	11	15
PSNR	Jacquin's	23.66	23.53	23.15
	Tong's	24.32	23.70	23.22
	Wang's	24.61	24.24	23.81
	Proposed	25.34	25.11	24.47
Bpp	Jacquin's	0.73	0.36	0.23
	Tong's	0.73	0.36	0.23
	Wang's	0.85	0.49	0.33
	Proposed	0.88	0.56	0.39
Time	Jacquin's	7.73	3.89	2.52
	Tong's	7.09	3.45	2.23
	Wang's	6.27	3.28	2.21
	Proposed	6.19	3.01	2.16

5.2. Fast Fractal Decoding Based on FSI. In the following part, we will estimate the performance of FSI. In order to describe the convergence speed conveniently, we define the following RMSE to describe the difference between each iteration and the final decoded image:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{f}_{ij}^{k} - \mathbf{f}_{ij}^{\mathrm{Decoded}}\right)^2}{M \times N}}, \quad k = 1, 2, \ldots, n, \tag{19}$$

where $\mathbf{f}^k$ and $\mathbf{f}^{\mathrm{Decoded}}$ are the kth iteration image and decoded image, respectively. Based on the fractal code, with a number of iterations, we first get the final decoded images as the reference image, and then we select the range-averaged image and FSI as the initial image, respectively. The same reconstruction method, described by (14), is adopted for both above two initial images. Figure 10(a) illustrates the RMSEs of each iteration with respect to the decoded image for Barbara image. Figure 10(b) illustrates the RMSEs with respect to the decoding time. We can see that FSI can provide smaller RSME all the time and achieve faster fractal decoding speed than the range-averaged image.

5.3. Hybrid Coding Method Based on the Proposed Fractal Coding Method. In this part, we first compare the proposed no-search fractal coding method with JPEG by compression ratio (CR) and PSNR. The error threshold T_1 in fractal coding is set to be 11 and the experimental results are listed in Table 3. We can see that, with almost the same decoded image quality, JPEG can provide higher CR than the fractal coding method for the whole image, but, for the blocks of 16 × 16 pixels, the fractal coding method can provide higher CR while maintaining almost the same decoded image quality as JPEG. Thus, we suppose that if the blocks of 16 × 16 pixels are encoded by the proposed fractal algorithm and the remaining blocks are encoded by JPEG, better performance can be obtained. The encoding process of the above hybrid coding method can be described as follows.

Step 1. Assign extra one bit to label the range block encoded by the fractal coding method.

Step 2. Partition the input image into range blocks of 16 × 16 pixels and encode them one by one. For arbitrary range block, if the collage error is smaller than T_1, encode the range block by the proposed fractal encoding method and label it; if not, partition the range block into four subblocks of 8 × 8 pixels.

Step 3. Encode all the remaining blocks of 8 × 8 pixels by JPEG encoder.

The corresponding decoding process can be described as follows.

Step 1. According to marking vector, decode the blocks of 16 × 16 pixels and 8 × 8 pixels with the fractal decoder and the JPEG decoder, respectively.

Step 2. Repeat Step 1 about ten times and the decoding process will converge to the final decoded image.

From the above analysis we know that, for the range blocks of 16 × 16 pixels, the fractal coding method can obtain

Table 3: Performance comparison between the proposed fractal coding method and JPEG while $T_1 = 11$.

Block size	Percentage	Fractal coding		JPEG	
		CR	PSNR (dB)	CR	PSNR (dB)
16 × 16	40.72%	70.62	32.85	57.46	32.86
8 × 8	41.19%	17.66	25.87	26.03	26.31
4 × 4 and 2 × 2	18.09%	4.36	20.00	16.55	19.82
Whole image	100%	14.16	25.11	29.55	25.14

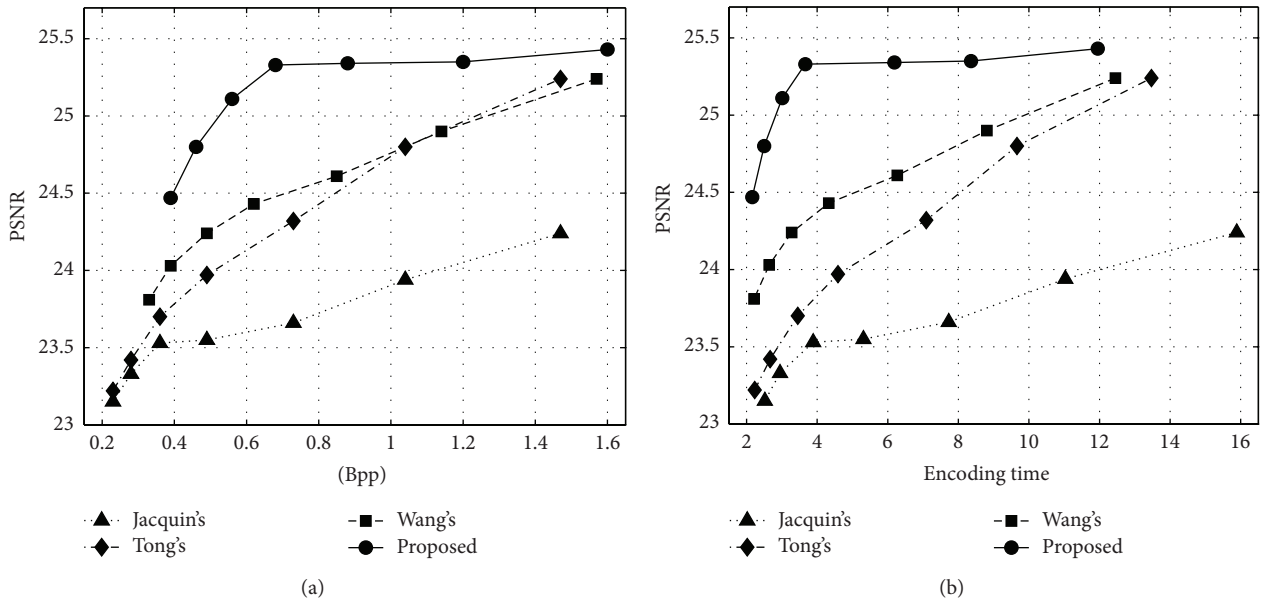


Figure 7: Comparison between the proposed method and the other three methods for Barbara image. (a) PSNR versus Bpp. (b) PSNR versus encoding time.

Figure 8: Decoded images of Barbara image; PSNR = 25.11 dB while $T_1 = 11$.

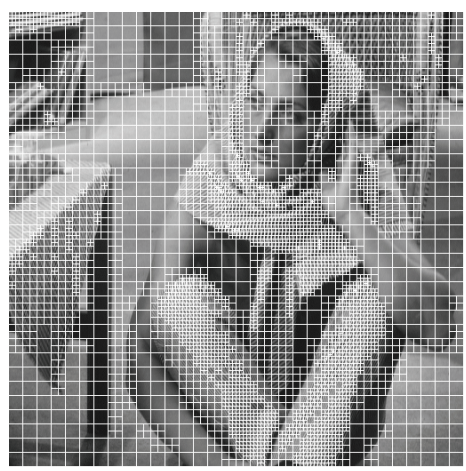


Figure 9: Quadtree structure for Barbara image while $T_1 = 11$.

good decoded image quality and high CR, simultaneously, but JPEG can only obtain either good decoded image quality or high CR. Thus, the decoded image quality can be improved by the local image encoded by the fractal coding method. Figure 11 illustrates the PSNR versus Bpp for the hybrid method and JPEG. It can be clearly seen that, with the same Bpp, the hybrid method can provide better PSNR than JPEG.

6. Conclusions

In order to reduce the matching error between the range block and its matching domain block, an improved gray-level

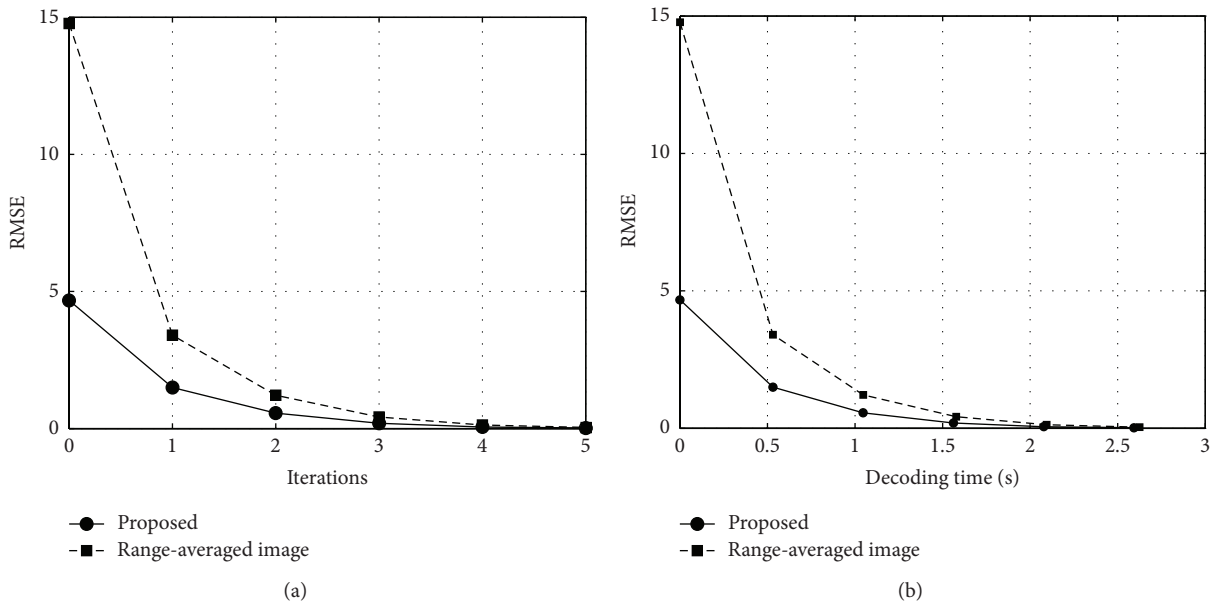


Figure 10: Comparison of the fractal decoding for the range-averaged image and FSI. (a) RMSE versus iterations. (b) RMSE versus decoding time.

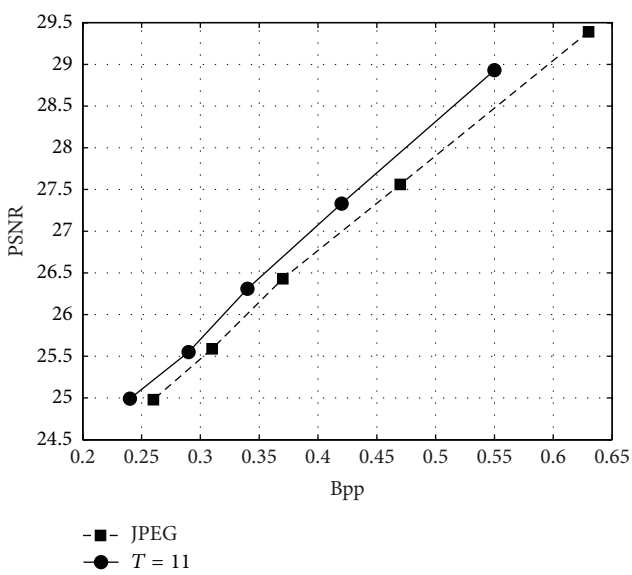


Figure 11: PSNR versus Bpp for the hybrid coding method and JPEG.

transform with a fitting surface is proposed in this paper. Firstly, the proposed fitting surface can approximate the range block better. Consequently, higher quality of the decoded image can be achieved with respect to the other three similar methods. Moreover, we adopt the same fitting surface for the range block and its matching domain block. This can save many bits and result in improving Bpp. In addition, due to smaller matching errors, more range blocks with larger size can be encoded and this can reduce the total number of blocks matching; that is, shorten the encoding time. Experimental results show that the proposed method can yield superior performance over the other three similar methods. Moreover, FSI can also accelerate the fractal decoding process. Finally, by combining the proposed fractal encoding method with JPEG, PSNR can be achieved higher than JPEG while maintaining the same Bpp.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the National 863 Program of China (Grant no. 2011AA110201) and the Fundamental Research Funds for the Central Universities (Grant no. 3132014090). The authors are grateful to the anonymous referees for their helpful comments and suggestions.

References

[1] M. F. Barnsley, *Fractal Everywhere*, Academic Press, New York, NY, USA, 1993.

[2] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions of Image Processing*, vol. 1, no. 1, pp. 18–30, 1992.

[3] B. Wohlberg and G. de Jager, "A review of the fractal image coding literature," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1716–1729, 1999.

[4] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal image denoising," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1560–1578, 2003.

[5] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal-wavelet image denoising revisited," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2669–2675, 2006.

[6] X. Wang and Z. Chen, "A fast fractal coding in application of image retrieval," *Fractals*, vol. 17, no. 4, pp. 441–450, 2009.

[7] X. Huang, Q. Zhang, and W. Liu, "A new method for image retrieval based on analyzing fractal coding characters," *Journal of Visual Communication and Image Representation*, vol. 24, no. 1, pp. 42–47, 2013.

[8] Y. C. Wee and H. J. Shin, "A novel fast fractal super resolution technique," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1537–1541, 2010.

[9] C. M. Lai, K. M. Lam, Y. H. Chan, and W. C. Siu, "An efficient fractal-based algorithm for image magnification," in *Proceeding of the International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP '04)*, pp. 571–574, Hong Kong, China, October 2004.

[10] J. Lu, Y. Zou, C. Yang, and L. Wang, "A robust fractal color image watermarking algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 638174, 12 pages, 2014.

[11] M. H. Pi, C. H. Li, and H. Li, "A novel fractal image watermarking," *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 488–498, 2006.

[12] F. Daraee and S. Mozaffari, "Watermarking in binary document images using fractal codes," *Pattern Recognition Letters*, vol. 35, pp. 120–129, 2014.

[13] Y. Fisher, *Fractal Image Compression: Theory and Application*, Springer, New York, NY, USA, 1994.

[14] B. Hurtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding of still images," in *Proceedings of the EOS/SPIE Visual Communications PACS Medical Applications*, pp. 397–408, Berlin, Berlin, 1993.

[15] C.-M. Lai, K.-M. Lam, and W.-C. Siu, "Improved searching scheme for fractal image coding," *Electronics Letters*, vol. 38, no. 25, pp. 1653–1654, 2002.

[16] C. J. He, X. Z. Xu, and J. Yang, "Fast fractal image encoding using one-norm of normalised block," *Chaos, Solitons and Fractals*, vol. 27, no. 5, pp. 1178–1186, 2006.

[17] Q. Wang, D. Liang, and S. Bi, "Fast fractal image encoding based on correlation information feature," in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP '10)*, pp. 540–543, Yantai, China, October 2010.

[18] S. Furao and O. Hasegawa, "A fast no search fractal image coding method," *Signal Processing: Image Communication*, vol. 19, no. 5, pp. 393–404, 2004.

[19] X. Wang and S. Wang, "An improved no-search fractal image coding method based on a modified gray-level transform," *Computers & Graphics*, vol. 32, no. 4, pp. 445–450, 2008.

[20] X. Y. Wang, Y. X. Wang, and J. J. Yun, "An improved no-search fractal image coding method based on a fitting plane," *Image and Vision Computing*, vol. 28, no. 8, pp. 1303–1308, 2010.

[21] R. Hamzaoui, "A new decoding algorithm for fractal image compression," *Electronics Letters*, vol. 32, no. 14, pp. 1273–1274, 1996.

[22] G. E. Øien and S. Lepsøy, "Fractal-based image coding with fast decoder convergence," *Signal Processing*, vol. 40, no. 1, pp. 105–117, 1994.

[23] H. S. Kang and S. D. Kim, "Fractal decoding algorithm for fast convergence," *Optical Engineering*, vol. 35, no. 11, pp. 3191–3198, 1996.

[24] Y. H. Moon, H. S. Kim, and J. H. Kim, "A fast fractal decoding algorithm based on the selection of an initial image," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 941–945, 2000.

[25] C. S. Tong and M. Pi, "Fast fractal image encoding based on adaptive search," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1269–1277, 2001.